
Analisis Perbandingan Performa Framework NestJS dan Lumen Pada Studi Kasus Aplikasi Berbasis REST API

Buce Trias Hanggara^{1*}, Muhammad Hasan Nasrullah², Djoko Pramono³

^{1,2,3}Universitas Brawijaya, Ilmu Komputer, Sistem Informasi, Jl Veteran Malang, Indonesia

***Email Korespondensi:**
Buce_trias@ub.ac.id

Abstrak

Sistem layanan informasi saat ini seringkali dikembangkan dengan arsitektur webservice berbasis Application Programming Interface (API). Terdapat banyak teknologi API yang bisa digunakan dan memiliki tingkat performa yang berbeda. NodeJS dan PHP memiliki framework khusus untuk pembuatan backend tersebut, yakni NestJS dan Lumen. Penelitian ini berfokus untuk membandingkan performa aplikasi dari sudut pandang response time, CPU, dan Memmory Usage. Metode yang digunakan adalah dengan menguji 8 layanan yang dibuat menggunakan masing-masing teknologi. Setelah melakukan pengujian signifikansi data, peneliti memberikan hasil perbandingan dimana kedua teknologi tidak memiliki perbedaan dalam kecepatan waktu respon, dimana keduanya hanya memiliki selisih 40ms atau 0,04 detik. Dalam aspek lain, terdapat perbedaan antara kedua teknologi dalam penggunaan CPU dengan selisih 6% dan memori dengan selisih 1%. NestJS lebih baik dalam efisiensi penggunaan CPU dan memori dibandingkan Lumen, berdasarkan hasil perhitungan rata-rata.

Kata Kunci: CPU; Lumen; Memory Usage; NestJS

Abstract

Information service systems are often developed using web service architectures based on Application Programming Interfaces (APIs). There are various API technologies available, each offering different performance levels. NodeJS and PHP have specific frameworks for backend development, namely NestJS and Lumen. This study focuses on comparing the performance of these applications from the perspective of response time, CPU, and memory usage. The method involves testing 8 services created using each technology. This research provides a comparative analysis showing that there is no significant difference in response time between the two technologies, with only a 40ms or 0.04-second difference. In other aspects, differences were found between the two technologies in terms of CPU usage, with a 6% difference, and memory usage, with a 1% difference. NestJS demonstrated better efficiency in CPU and memory usage compared to Lumen, based on average calculations.

Keywords: CPU; Lumen; Memory Usage; NestJS

1. Pendahuluan

Proses jalannya bisnis telah didominasi penggunaan proses bisnis yang dibagi ke dalam layanan-layanan yang spesifik. Untuk mendukung hal tersebut, teknologi informasi juga beradaptasi dengan adanya pengembangan teknologi berbasis layanan, atau bisa disebut *Service Oriented Architecture* (SOA). Pengembangan *Backend* aplikasi dengan arsitektur tersebut, umumnya didasari pada penggunaan *Application Programming Interface* (API) (Hasanuddin et al., 2022). Sebelum REST, SOAP merupakan API yang paling umum digunakan. Namun sejak 2000-an, REST atau RESTful API merupakan API yang paling umum digunakan saat ini (Laksono et al., 2024; Putra et al., 2019; Team, 2023). Oleh karena itu, REST API disini dipilih sebagai Webservice yang akan diteliti.

Representational State Transfer (REST) adalah arsitektur komunikasi yang beroperasi di atas protokol HTTP dan sering digunakan dalam pengembangan sistem informasi (Luky Mulana et al., 2022). REST dapat dikembangkan menggunakan berbagai bahasa pemrograman dan framework, yang menyebabkan variasi performa REST API tergantung pada teknologi yang digunakan. Beberapa aspek yang mempengaruhi performa dan dapat diteliti termasuk diantaranya adalah penggunaan CPU, penggunaan RAM, dan waktu respons (Choirudin et al., 2019). Waktu respons yang cepat menjadi penting ketika aplikasi harus menangani banyak permintaan dari klien, sementara efisiensi penggunaan CPU dan RAM penting untuk mengoptimalkan sumber daya dan biaya selama operasional aplikasi (Rompis, 2018).

Ada beberapa teknologi atau framework yang sering digunakan untuk pengembangan REST API. PHP, yang dirancang untuk pengembangan aplikasi web, memiliki framework seperti Lumen yang digunakan untuk mengembangkan API microservices dengan cepat (Aqshal Marta Yudha et al., 2022) Berdasarkan data dari NPM (Node Package Manager), NestJS telah diunduh sebanyak 3.184.614 kali dalam satu minggu, sementara berdasarkan Packagist, Lumen telah diunduh sebanyak 19.643.020 kali. Berdasarkan jumlah unduhan tersebut, penelitian ini akan membandingkan performa REST API yang dibangun menggunakan framework microservices PHP dan NestJS.

Pada penelitian yang sudah dilakukan sebelumnya menunjukkan hasil performa antara teknologi Javascript dan PHP dengan hasil ExpressJS memiliki response time yang lebih cepat dibandingkan Codeigniter. Namun Codeigniter menggunakan resource yang lebih efektif daripada ExpressJS. Peneliti pada penelitian tersebut memberikan saran yang menjadi bahan pertimbangan dasar penelitian ini dilakukan yaitu disarankan untuk melakukan percobaan dengan framework backend yang lain untuk pengembangan RESTful API Mulana (2022). Oleh karena itu, peneliti memilih microframework Lumen dari PHP dan microframework NestJS dari Javascript atau NodeJS. Perbandingan ini lebih setara karena membandingkan microframework dari masing-masing teknologi

Berdasarkan studi kasus tersebut, penelitian akan dilakukan untuk mengetahui performa yang lebih baik antara aplikasi NodeJS dan PHP dengan membandingkan variabel *response time*, CPU, dan *memory usage*. Harapan dari penelitian ini dapat digunakan baik dari akademisi maupun praktisi untuk dapat menggunakan teknologi yang tepat guna sesuai dengan kebutuhan masing-masing teknologi.

1.1 Arsitektur Web Service

Web services merupakan sistem yang dikembangkan untuk mendukung sistem terdistribusi yang memungkinkan dibangun dengan infrastruktur yang berbeda-beda. *Web services* dapat berkomunikasi dengan *web services* yang lain (Papazoglou et al., 2004).

Web services berjalan dengan menggunakan protokol HTTP. *Web services* dikembangkan pada web untuk menyediakan informasi antar sistem pada suatu jaringan. Informasi di bentuk dengan format XML ataupun JSON untuk agar dapat digunakan oleh sistem yang berbeda. Setiap *web services* dapat diakses oleh pengguna ataupun sistem lain dengan menggunakan fungsi dan *method* yang telah disediakan oleh server. *Web Services* sangat memungkinkan untuk dikembangkan dengan bahasa pemrograman yang berbeda-beda (Paramartha et al., 2017).

1.2 Lumen

Lumen yang dimaksud pada penelitian ini adalah *framework microservice* yang merupakan bagian dari *framework* Laravel. Laravel sendiri merupakan *framework* yang berbasis bahasa pemrograman PHP (Lumen, n.d.). Penggunaan Laravel pada umumnya digunakan untuk membuat website secara keseluruhan. Berbeda dengan Laravel, Lumen berfokus pada library yang digunakan untuk mengembangkan RESTful API dengan beberapa fitur yang didukung Laravel. Adapun beberapa fitur tersebut diantaranya fitur *validation*, *middleware*, Eloquent ORM, *routing* dan lain sebagainya (Aqshal Marta Yudha et al., 2022).

1.3 NestJS

NestJS adalah framework JavaScript open-source yang dirancang untuk membangun aplikasi REST. Teknologi ini merupakan hasil pengembangan dari framework ExpressJS atau Fastify, dengan penambahan fitur-fitur modular baru (Muhamad Saepuloh et al., 2022) .

NestJS sepenuhnya didukung oleh TypeScript dan mengintegrasikan konsep OOP (*Object Oriented Programming*), FP (*Functional Programming*), serta FRP (*Functional Reactive Programming*). Pihak NestJS mengklaim bahwa pengembangan dengan teknologi ini dapat dilakukan lebih cepat, mudah dikembangkan, dan kodenya dapat diuji kesesuaiannya dengan baik(NestJS, n.d.).

1.4 Paired Sample T-Test

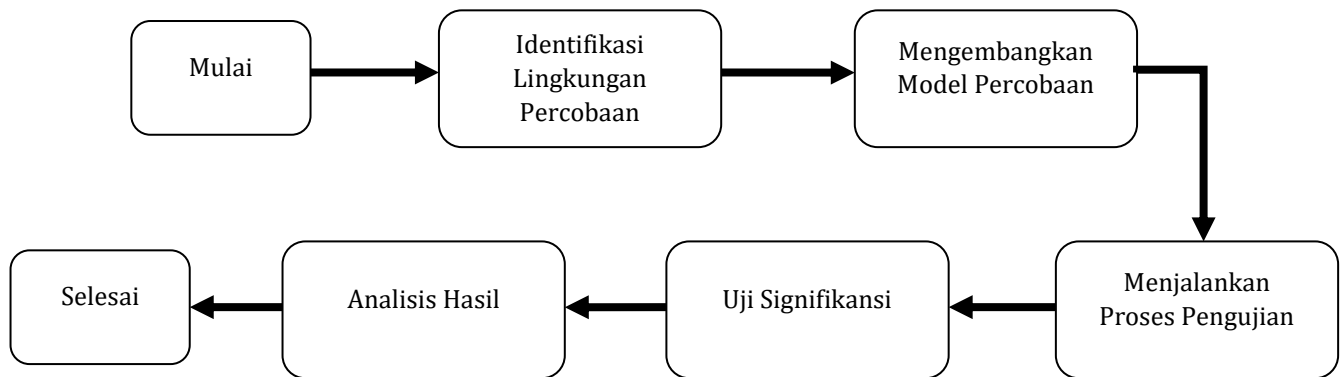
Paired Sample T-Test merupakan pengujian signifikansi data dari 2 sample yang berpasangan yang mengalami 2 pengukuran yang berbeda seperti Subjek A dengan metode A1 dan metode A2 (Ravis, dkk., 2019). Dalam kondisi penelitian ini setiap layanan akan diperbandingkan dengan 2 teknologi yang berbeda. Tujuan dari dilakukan tes ini adalah untuk mengetahui apakah 2 teknologi yang dibandingkan dalam memiliki perbedaan yang signifikan atau tidak. Kriteria yang telah ditentukan pada pengujian ini adalah memenuhi syarat signifikansi dengan nilai Sig. kurang dari 0.05 dapat dilihat pada Tabel 1.

Table 1. Syarat Paired Sample T-Test

Syarat	Signifikansi
p-value < 0.05	Terdapat perbedaan
p-value > 0.05	Tidak terdapat perbedaan

2. Metode Penelitian

Penelitian ini termasuk dalam penelitian analitik. Dengan 5 tahap penelitian yaitu identifikasi lingkungan percobaan, mengembangkan model percobaan, menjalankan proses pengujian, melakukan uji signifikansi dan menganalisa hasil pengujian. Gambar 1 menjelaskan terkait diagram alir metodologi penelitian tersebut



Gambar 1. Diagram alir metodologi penelitian

Tahapan pertama yakni identifikasi lingkungan percobaan yaitu pada Virtual Private Server dengan spesifikasi memori 1GB dan CPU 1 Core pada OS Ubuntu 22. Database yang digunakan adalah MySQL. Selanjutnya pengembangan model percobaan, dimana pada kasus uji ini dibuat ke dalam 8 bentuk layanan REST. HTTP method yang digunakan untuk layanan tersebut adalah POST, GET, PUT dan DELETE.

Selanjutnya, eksperimen dilakukan dengan menggunakan 100-500 thread yang diulang sebanyak lima kali untuk setiap jumlah thread. Pengumpulan data akan menggunakan alat JMeter dengan plugin Perfmon dengan

menangkap hasil performa berdasarkan kecepatan waktu respon, penggunaan CPU dan penggunaan RAM. Data akan berupa nilai dari setiap aspek pada setiap pengujian. Pengujian akan dilakukan sebanyak 5 kali dengan adanya peningkatan jumlah permintaan yang harus diterima oleh server yang menggunakan masing-masing framework. Dengan adanya alat tersebut, maka data yang dihasilkan akan lebih akurat.

Setelah itu, data hasil eksperimen dianalisis dengan menghitung rata-rata untuk setiap jumlah thread dan dilakukan uji statistik menggunakan metode Komolgorov-Smirnov dan Paired Sample T-Test untuk membuktikan apakah terdapat hubungan yang signifikan dari hasil pengujian tersebut. Hasil pengujian kemudian di analisis menggunakan kelimuan dan penelitian terdahulu sebagai acuan.

Untuk melakukan pengujian tersebut, dibuatlah delapan layanan yang akan dibuat menggunakan dua teknologi yang berbeda. Daftar layanan tersebut dapat dilihat pada Tabel 2.

Table 2. Layanan yang digunakan pada kasus uji

No	Method	Nama	Endpoint	Keterangan
1	POST	Login	/api/login	Layanan untuk autentikasi admin
2	GET	Get All Order	/api/order	Layanan untuk melihat semua pesanan
3	GET	Get Detail Order	/api/order/{id}	Layanan untuk melihat detail pesanan
4	POST	Post Order	/api/order	Layanan untuk menambah pesanan
5	GET	Get All Service	/api/service	Layanan untuk melihat semua jenis layanan
6	POST	Post Service	/api/service	Layanan untuk menambah jenis layanan
7	PUT	Put Service	/api/service/{id}	Layanan untuk mengubah jenis layanan
8	DELETE	Delete Service	/api/service/{id}	Layanan untuk menghapus jenis layanan

3. Hasil dan Pembahasan

Bagian hasil dan pembahasan ini akan membahas terkait pengujian normalitas, uji signifikansi, serta analisis terkait hasil pengujian yang didapatkan. Uji normalitas dilakukan untuk mencari apakah sebaran data yang diambil terdistribusi normal atau tidak. Jika distribusi data dinyatakan normal, maka pengujian parametrik bisa dilakukan. Namun jika distribusi tidak normal, maka akan dilakukan pengujian non parametrik untuk menggali signifikansi data.

Berikut ini adalah contoh data yang sudah didapat dari pengambilan data menggunakan JMeter. Data yang dijadikan sample ini adalah salah satu contoh hasil data pengujian dari beberapa data pengujian yang ada. Adapun keseluruhan data yang didapat adalah 24 kelompok data, dimana dihitung dari tiga (3) variabel (response time, CPU, memori) dikalikan dengan delapan (8) total layanan yang dibuat. Tabel 3 berikut, menggambarkan data waktu response di layanan login.

Table 3. Hasil data pengujian Response Time untuk Layanan Login

No	User (Threads)	NestJS (NodeJS)	Lumen (PHP)
1	100	3503,7	3617,1
2	200	7243,3	7271,4
3	300	10984,3	11051,4
4	400	14508,7	15375,0
5	500	18128,2	19260,6

3.1 Uji Normalitas

Uji normalitas dilakukan dengan cara menguji keseluruhan data yang diambil pada kasus uji NestJS dan Lumen. Data yang digunakan adalah rata-rata *response time*, CPU, dan *memory usage* pada saat dilakukan seluruh kasus uji. Kasus uji yang dimaksud pada tahapan pengujian adalah layanan: Login, Get All Order, Get Detail

Order, Post Order, Get All Service, Post Service, Put Service, dan Delete Service. Tabel 4 berikut adalah salah satu contoh sampling dari uji normalitas waktu respons untuk penggunaan Lumen.

Berdasarkan keseluruhan hasil pengujian, untuk keseluruhan layanan, baik menggunakan Lumen maupun NestJS, begitupun untuk seluruh variabel yakni *response time*, CPU, dan *memory usage*, hasil nilai menyatakan bahwa seluruh data terdistribusi normal. Oleh sebab itu, pengujian signifikansi selanjutnya dapat dilakukan secara parametrik menggunakan *Paired Sample T-Test*.

Table 4. Uji Normalitas response time pada Lumen

<i>Layanan</i>	<i>Rata-rata (ms)</i>	<i>Sig.</i>
Login	11315,1	.200
Get All Order	2734,9	.200
Get Detail Order	1714,9	.200
Post Order	2035,2	.200
Get All Service	1860,5	.200
Post Service	1637,4	.120
Put Service	1610,2	.115
Delete Service	1676,7	.160

3.2 Uji Signifikansi

Uji Signifikansi dilakukan dengan cara memasukkan keseluruhan data rata-rata masing-masing variabel yang diukur kedalam alat uji statistik. Berikut merupakan hasil perhitungan p-value, dimana nilai tersebut menunjukkan adanya perbedaan yang signifikan maupun tidak, antara variable uji pada studi kasus di Lumen dan NestJS. Tabel 5 menunjukkan signifikansi waktu respon, Tabel 6 menunjukkan signifikansi penggunaan CPU, dan Tabel 7 menunjukkan signifikansi pada penggunaan memori.

Table 5. Uji Signifikansi waktu respon

<i>Layanan</i>	<i>p-Value</i>
Login	.130
Get All Order	.010
Get Detail Order	.159
Post Order	.085
Get All Service	.023
Post Service	.106
Put Service	.272
Delete Service	.124

Untuk response time sendiri, data menunjukkan bahwa tidak ada perbedaan yang signifikan diantara dua teknologi yang digunakan. Seperti yang sudah dijelaskan pada bagian teori sebelumnya, kriteria yang telah ditentukan pada pengujian ini adalah memenuhi syarat signifikansi dengan nilai p-value kurang dari 0.05. Sedangkan pada tabel tersebut menunjukkan bahwa 5 dari 8 layanan menunjukkan nilai diatas 0.05. Hal itu menunjukkan bahwa pada penelitian ini, response time antara NestJS dan Lumen tidak memiliki perbedaan yang signifikan.

Table 6. Uji Signifikansi penggunaan CPU

<i>Layanan</i>	<i>p-Value</i>
Login	.019
Get All Order	.033

Get Detail Order	.001
Post Order	.001
Get All Service	.001
Post Service	.001
Put Service	.113
Delete Service	.001

Pada kasus penggunaan CPU, berdasarkan hasil p-value yang didapati pada tiap pengujian, didapatkan bahwa 7 dari 8 layanan memiliki nilai dibawah 0.05. Hal itu berarti terdapat perbedaan nilai yang signifikan pada penggunaan CPU.

Table 7. Uji Signifikansi penggunaan CPU

Layanan	p-Value
Login	.000
Get All Order	.137
Get Detail Order	.416
Post Order	.009
Get All Service	.000
Post Service	.000
Put Service	.000
Delete Service	.000

Sama halnya pada kasus *memory usage*, berdasarkan hasil p-value yang didapati pada tiap pengujian, didapatkan bahwa 6 dari 8 layanan memiliki nilai dibawah 0.05. Hal itu berarti terdapat perbedaan nilai yang signifikan pada penggunaan Memori antara teknologi yang dibuat menggunakan NestJS dan Lumen..

3.3 Pembahasan

Dari hasil uji signifikansi yang telah dilakukan sebelumnya, didapatkan perbedaan yang tidak terlalu signifikan antara penggunaan Lumen maupun NestJS. Namun, kita dapat tetap menilai perbedaan tersebut seperti yang dapat dilihat pada tabel 8 berikut.

Table 8. Perbandingan rerata waktu respon (dalam ms)

Layanan	NestJS	Lumen	Lebih unggul
Login	10873,6	11315,1	NestJS
Get All Order	3244,0	2734,9	Lumen
Get Detail Order	1470,7	1714,9	NestJS
Post Order	1574,5	2035,2	NestJS
Get All Service	2605,5	1860,5	Lumen
Post Service	1470,3	1637,4	NestJS
Put Service	1559,5	1610,2	NestJS
Delete Service	1434,6	1676,7	NestJS

Setelah penelitian dilakukan, berdasarkan hasil uji signifikansi, 6 dari 8 layanan tidak menunjukkan perbedaan kecepatan waktu respons antara kedua teknologi. Namun, 2 dari 8 layanan menunjukkan perbedaan yang signifikan. Jika fitur yang dikembangkan melibatkan pengambilan data dalam jumlah besar dari database, lebih baik menggunakan Lumen dengan EloquentORM yang sudah terintegrasi penuh. Sebaliknya, jika fitur melibatkan komputasi tinggi seperti perhitungan atau pengambilan data spesifik dengan ukuran respons kecil, maka NestJS dengan arsitektur non-blocking I/O single thread lebih disarankan (Prayogi et al., 2020). Secara keseluruhan, mayoritas layanan menunjukkan bahwa NestJS dan Lumen tidak memiliki perbedaan waktu respons yang signifikan, dengan rata-rata selisih 0,04 detik atau 40 ms.

Berikutnya, untuk penggunaan CPU, pengujian signifikansi menunjukkan adanya perbedaan yang signifikan. Dari hasil perbandingan rata-rata penggunaan CPU yang dapat dilihat pada Tabel 9, NestJS mendominasi

keunggulan penggunaan CPU yang lebih rendah, dibandingkan Lumen. Seluruh layanan, kecuali Get All Service menunjukkan bahwa penggunaan CPU pada teknologi berbasis NestJS lebih efisien.

Hasil ini sejalan dengan penelitian yang dilakukan oleh Hadinata et al. (2022). Dalam penelitian dilakukan perbandingan performa antara framework ExpressJS dari NodeJS dan Laravel dari PHP. Hasil dari penelitian ini, penggunaan rata-rata CPU ExpressJS (NodeJS) lebih efisien dalam penggunaan CPU daripada Laravel (PHP) dengan selisih 3%(Hadinata et al., 2024).

Table 9. Perbandingan rerata penggunaan CPU

<i>Layanan</i>	<i>NestJS</i>	<i>Lumen</i>	<i>Lebih unggul</i>
Login	98,3	98,8	NestJS
Get All Order	44,3	68,9	NestJS
Get Detail Order	28,5	52,1	NestJS
Post Order	41,6	59,5	NestJS
Get All Service	62,6	30,6	Lumen
Post Service	32,5	49,9	NestJS
Put Service	41,0	49,3	NestJS
Delete Service	37,6	50,2	NestJS

Pada pengujian yang terakhir, didapatkan bahwa NestJS kembali memiliki catatan penggunaan memori yang lebih sedikit jika dibandingkan dengan teknologi berbasis Lumen. Mayoritas layanan menggunakan memori yang lebih rendah, kecuali pada proses Get Detail Order dan Get All Services. Hasil ini sejalan dengan penelitian yang dilakukan oleh (Greiff et al., 2019). Dalam penelitian tersebut dilakukan perbandingan penggunaan memori antara ExpressJS (NodeJS) dan Symfony (PHP). Pada hasil dari penelitian tersebut, ExpressJS (NodeJS) menggunakan 10,86% memori sedangkan Symfony (PHP) menggunakan 19,02% memori. Berdasarkan hasil penelitian tersebut, ExpressJS (NodeJS) lebih baik dalam penggunaan memori. Perbandingan rata-rata penggunaan memori tersebut dapat dilihat pada Tabel 10.

Table 10. Perbandingan rerata penggunaan memori

<i>Layanan</i>	<i>NestJS</i>	<i>Lumen</i>	<i>Lebih unggul</i>
Login	40,1	45,0	NestJS
Get All Order	76,7	77,5	NestJS
Get Detail Order	75,1	75,0	Lumen
Post Order	62,3	62,6	NestJS
Get All Service	64,6	70,1	Lumen
Post Service	71,9	63,4	NestJS
Put Service	65,4	68,0	NestJS
Delete Service	67,8	65,5	NestJS

4. Kesimpulan

Setelah melakukan pengujian pada bab sebelumnya terhadap framework NestJS dan Lumen menggunakan JMeter, peneliti menyimpulkan bahwa secara umum tidak terdapat perbedaan signifikan antara kedua framework dalam hal kecepatan waktu respon. Pengujian signifikansi menunjukkan bahwa mayoritas layanan tidak memiliki perbedaan signifikan terkait kecepatan waktu respon. Namun, terdapat dua layanan, yaitu "get all order" dan "get all service," yang menunjukkan perbedaan signifikan, di mana Lumen lebih cepat dalam

mengambil data dalam jumlah besar. Hal ini disebabkan oleh penggunaan library yang terhubung dengan database dan proses pengambilan data dalam jumlah besar, yang lebih efisien pada Lumen.

Pengujian selanjutnya berkaitan dengan efisiensi penggunaan CPU dan memori. Hasil pengujian signifikansi menunjukkan perbedaan performa antara kedua teknologi. Berdasarkan perhitungan rata-rata, layanan yang dikembangkan menggunakan NestJS cenderung menggunakan sumber daya yang lebih sedikit dibandingkan dengan Lumen. Oleh karena itu, peneliti menyimpulkan bahwa NestJS lebih efisien dalam penggunaan CPU dengan selisih 6% dibandingkan dengan Lumen. NestJS juga lebih efisien dalam penggunaan memori dengan selisih 1% dibandingkan dengan Lumen.

Saran untuk penelitian selanjutnya, dapat dikaji lebih lanjut terkait penggunaan framework lain yang diharapkan dapat memberikan bukti yang signifikan pada variabel *response time*. Penelitian lain juga memungkinkan adanya teknik pengambilan data selain menggunakan tools JMeter seperti yang digunakan pada penelitian ini.

5. Referensi

- Aqshal Marta Yudha, & Andhik Budi Cahyono. (2022). Pengembangan Back End Menggunakan Laravel Lumen (Studi Kasus Teknologi.id Event). *Automata (Ajang Unjuk Tugas Akhir Oleh Mahasiswa Informatika)*, 3(2).
- Choirudin, R., & Adil, A. (2019). Implementasi Rest Api Web Service dalam Membangun Aplikasi Multiplatform untuk Usaha Jasa. *MATRIK: Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, 18(2), 284–293. doi: 10.30812/matrik.v18i2.407
- Greiff, M., & Johansson, A. (2019). *Symfony vs Express: A Server-Side Framework Comparison* (p. 49).
- Hadinata, W., & Stianingsih, L. (2024). ANALISIS PERBANDINGAN PERFORMA RESTFULL API ANTARA EXPRESS.JS DENGAN LARAVEL FRAMEWORK. *Jurnal Informatika Dan Teknik Elektro Terapan*, 12(1). doi: 10.23960/jitet.v12i1.3845
- Hasanuddin, Asgar, H., & Hartono, B. (2022). RANCANG BANGUN REST API APLIKASI WESHARE SEBAGAI UPAYA MEMPERMUDAH PELAYANAN DONASI KEMANUSIAAN. *Jurnal Informatika Teknologi Dan Sains*, 4(1), 8–14. doi: 10.51401/jinteks.v4i1.1474
- Laksono, M. A., Kautsar, I. A., & Setiawan, H. (2024). Implementasi Payment Gateway pada Platform Freelance Digital Menggunakan Rest API. *SMATIKA JURNAL*, 14(01), 135–145. doi: 10.32664/smatika.v14i01.1227
- Luky Mulana, Kamal Prihandani, & Adhi Rizal. (2022). Analisis Perbandingan Kinerja Framework Codeigniter Dengan Express.js Pada Server RESTful Api. *JURNAL ILMIAH WAHANA PENDIDIKAN (JIWP)*, 8(16). doi: <https://doi.org/10.5281/zenodo.7067707>
- Lumen. (n.d.). *Dokumentasi Lumen*. Retrieved from <https://lumen.laravel.com/docs/11.x>
- Muhamad Saepuloh, A., & Ginting, S. (2022). PERANCANGAN SISTEM INFORMASI MANAJEMEN PROYEK DENGAN MENGGUNAKAN SOFTWARE NEST.JS BERBASIS WEB DI PT. MITRA PAJAKKU. *INFOKOM (Informatika & Komputer)*, 10(1), 1–9. doi: 10.56689/infokom.v10i1.818
- NestJS. (n.d.). *Website NestJS*. Retrieved from <https://docs.nestjs.com/>
- Papazoglou, M. P., & Dubray, J. (2004). *A Survei Of Web Services Technologies. Departmen of Information and Communication technology*.
- Paramartha, A. A. G. Y., Suryaningsih, G. K., & Aryanto, K. Y. E. (2017). IMPLEMENTASI WEB SERVICE PADA SISTEM PENGINDEKSAN DAN PENCARIAN DOKUMEN TUGAS AKHIR, SKRIPSI, DAN PRAKTIK KERJA LAPANGAN. *JST (Jurnal Sains Dan Teknologi)*, 5(2). doi: 10.23887/jstundiksha.v5i2.8813

- Prayogi, A. A., Niswar, M., Indrabayu, & Rijal, M. (2020). Design and Implementation of REST API for Academic Information System. *IOP Conference Series: Materials Science and Engineering*, 875(1), 012047. doi: 10.1088/1757-899X/875/1/012047
- Putra, G. Z., Mahendra, N. B., Kusuma, M. B. I., & Satriawan, G. A. (2019). Aplikasi Deteksi Gempa Secara Realtime Berbasis Mobile di Indonesia. *J-INTECH*, 7(02), 135–139. doi: 10.32664/j-intech.v7i02.440
- Rompis, A. C. (2018). Perbandingan Performa Kinerja Node.js, PHP, dan Python dalam Aplikasi REST. *CogITO Smart Journal*, 4(1), 160. doi: 10.31154/cogito.v4i1.92.160-170
- Team, T. P. (2023, April 26). *What Is a REST API? Examples, Uses, and Challenges*. Postman. Retrieved from <https://blog.postman.com/rest-api-examples/>